# Detecting and Tracking Dynamic Clusters of Spatial Events

Natalia Andrienko[1,2,3], Gennady Andrienko[1,2,3], Georg Fuchs[1,3], Hendrik Stange[1]

[1] Fraunhofer Institute IAIS; [2] City University London; [3] University of Bonn

**ABSTRACT**

We present a work in progress on developing a tool supporting real-time detection of significant clusters of spatial events and observing their evolution. The tool consists of an incremental stream clustering algorithm and coordinated map and timeline displays showing current situation and cluster evolution.

## 1 PROBLEM STATEMENT

Spatial events are physical or abstract entities with limited existence times and particular locations in space. A spatial event is characterized by its start and end times (which may coincide), spatial coordinates, and, possibly, some thematic attributes. We assume that occurrences of spatial events are registered, e.g., by sensors, and corresponding data records are immediately sent to a server. The resulting data stream needs to be monitored.

We consider such monitoring scenarios in which singular events are not significant and do not require attention of the observer, but occurrences of several events closely in space and time (i.e., spatio-temporal clusters of events) may be significant. For example, moving vehicles may emit low speed events when their speed drops below a certain threshold. It is neither feasible nor meaningful to attend to every such event. However, a spatio-temporal cluster of low speed events emitted by several cars may indicate a traffic jam and therefore require observer's attention. Furthermore, when a cluster emerges, the observer may need to trace its evolution, i.e., changes in the number of events, number of vehicles involved, spatial location, shape, and extent.

Our objective is to support the observer in detecting the emergence and tracking the evolution of spatial event clusters in real time. To reduce the observer's workload, clusters need to be detected automatically. Detected clusters are presented on visual displays. The observer should be able to see both the current situation, i.e., where what clusters exist at the present moment, and the evolution history of the clusters.

## 2 APPROACH

Automated on-line detection and tracking of spatio-temporal clusters of spatial events requires an appropriate data stream clustering algorithm. The algorithms proposed in data mining literature are not quite suitable for our purpose. Their main purpose is effective summarization and compact representation of an incoming data stream when individual data records are too numerous to be stored. It is often assumed that the data distribution is the same throughout the stream. A classical example is CluStream framework [1]. First, an initial portion of a data stream is partitioned into a user-given number of clusters, which are represented by special statistical summaries. Then, the main task of the algorithm is to maintain these clusters, i.e., for each new record, choose the appropriate cluster and update the statistical summary. Algorithms like this are not suitable for applications where clusters emerge, evolve, and disappear.

Closer to our needs is DenStream [2], which adapts the idea of density-based clustering to data stream settings. The algorithm groups incoming data into spherical micro-clusters. Dense micro-clusters are stored in a summarized form for further offline processing, in particular, for obtaining clusters of arbitrary shapes by applying the DBSCAN algorithm to the micro-clusters. While the idea of online generation of micro-clusters can work for us, our problem settings exclude offline post-processing.

In our approach, we also organize incoming events into micro-clusters. A micro-cluster consists of events fitting within a circle with the radius not exceeding a user-specified threshold $R$, called maximal neighborhood radius. When a new event comes, the algorithm checks whether it fits in one of existing micro-clusters, i.e., whether its distance to the center of the micro-cluster does not exceed $R$. If so, the event is added to this micro-cluster, and the position of the center is updated. If not, a new micro-cluster is created. It consists of only this event, the spatial position of which is taken as the micro-cluster center. For effective search of candidate micro-clusters for including new events, we use a spatial index structure [3].

The algorithm does not keep all events in the memory but only recent events, specifically, events having occurred within the time interval $[t\text{-}t_{gap}, t]$, where $t$ is the current time moment and $t_{gap}$ is a user-specified maximal allowed time gap between consecutive events in a cluster. Events becoming older than $t\text{-}t_{gap}$ are erased from the micro-clusters. When new events are added to a micro-cluster and when old events are erased, the resulting states of the micro-cluster are recorded in its history. A state is represented by a record specifying the time moment when the state was achieved (i.e., the current moment), the current position of the micro-cluster center, the radius, the count of the member events at the current moment, and the total (cumulative) number of the member events since the emergence of the micro-cluster. Optionally, a state record may also contain the spatial convex hull of the current member events.

When, after erasing old events, no events remain in a micro-cluster, it stops existing, i.e., it is removed from the main memory. If the total count of member events is less than a user-specified minimum $N_{min}$, the micro-cluster is simply forgotten; otherwise, the history of the micro-cluster is offloaded to external storage.

Full spatio-temporal clusters of arbitrary shapes are obtained by merging neighboring micro-clusters. We introduce a concept of connecting event, which is an event located sufficiently close to centers of two or more micro-clusters. By default, "sufficiently close" means that the distance does not exceed the maximal neighborhood radius $R$, but the user may also specify a special connection distance threshold $R_c$. When a pair of micro-clusters has at least $k$ connecting events, where $k$ is a user-specified parameter, these micro-clusters can be merged into a larger cluster. A similar idea is employed in algorithm AING [4], where analogs of micro-clusters are treated as graph nodes. When a connecting data point appears, an edge is created between the respective nodes.

In our algorithm, as soon as two micro-clusters become $k$-connected, they are merged, which means that a macro-cluster is created in which the micro-clusters are members. A macro-cluster may include an arbitrary number of micro-clusters provided that each of them is $k$-connected to some other member micro-cluster. Hence, macro-clusters may have arbitrary shapes and sizes.

In event clustering, not only the spatio-temporal positions but also thematic attributes can be taken into account. For example, low speed events emitted by moving vehicles may include an attribute 'movement direction', which can be used in clustering so

IEEE Symposium on Visual Analytics Science and Technology 2014
November 9-14, Paris, France
978-1-4799-6227-3/14/$31.00 ©2014 IEEE

that only events with similar directions may be put together (the user needs to specify the maximal allowed difference in the directions). Another extension is accounting for the event sources (e.g., the vehicles emitting low speed events). For each micro- and macro-cluster, the algorithm can maintain a list of distinct event sources. Clusters where all events come from a single source or from too few distinct sources may be disregarded as insignificant.

During the stream observation process, clusters (macro-clusters and isolated micro-clusters) where the total event counts are not less than $N_{min}$ are visually presented to the observer on two coordinated dynamic views: a map and a timeline display. Both views show clusters existing during the time interval $[t, \Delta t]$, where $t$ is the current time moment and $\Delta t$ is a chosen time horizon of the observation. The map shows the spatial positions, extents, and shapes of the clusters. The timeline display shows the evolution of the clusters, i.e., changes in the number of events, spatial extent, density, and/or other characteristics of the clusters. Both displays are updated when changes of the shown clusters occur and when new clusters become significant in terms of the number of member events and, optionally, the number of distinct sources.

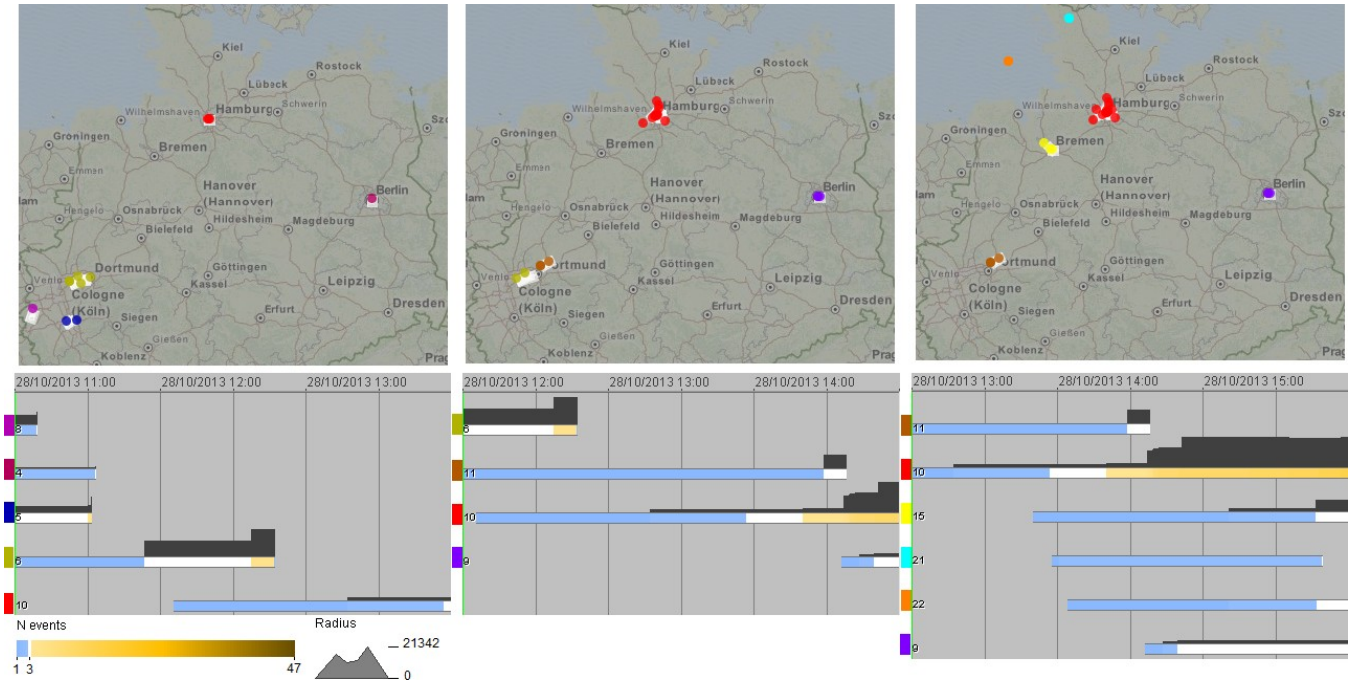


Figure 1. Emergence and evolution of spatio-temporal clusters of georeferenced tweets related to a hurricane on October 28, 2013.

## 3 EXAMPLE

Figure 1 illustrates a possible application scenario for event stream observation. For early detection and monitoring of impacts of disastrous and extraordinary events, a civil protection agency utilizes a stream of georeferenced tweets containing specific tags and keywords of interest (e.g., "storm", "hurricane", etc.). While considering every such tweet as an indication of a really happening event would result in very many false alarms, spatio-temporal clusters of tweets may be significant. Our example uses real georeferenced tweets posted on the territory of Germany in October 27-28, 2013, when hurricane Christian happened.

Taking into account the spatial scale of the analysis (the whole territory of Germany) and the sparseness of the relevant tweet events, the maximal neighborhood radius $R$ has been set to 15 km and the minimal significant number of events in a cluster $N_{min}$ to 3. The minimal number of connecting events for merging micro-clusters $k$ has been set to 1.

Three pairs of screenshots of the map and timeline display correspond to situations at 14:00, 15:00, and 16:00 on October 28 with a temporal horizon of 3 hours. In the map, the detected significant clusters are represented by their convex hulls (white polygons with thick boundary lines for better visibility), and their member events by colored circles. The events that are not members of the significant clusters are not shown, to avoid distracting observer's attention. In the timeline display, significant clusters are represented by bars. Bar coloring can represent some of the cluster characteristics, in our example, the number of member tweet events. Other characteristics can be additionally represented by heights and coloring of polygons attached above and/or below the bars. In our example, the heights of the polygons above the bars show the cluster radii at different time moments. Hence, the timeline view allows the observer to see how significant clusters evolved during the observation time horizon.


## ACKNOWLEDGEMENT

This work is supported by EU in project INSIGHT "Intelligent Synthesis and Real-time Response using Massive Streaming of Heterogeneous Data" (http://insight-ict.eu/).


## REFERENCES


[1] Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams, in: *Proc. 29th Int. Conf. Very Large Data Bases*, Berlin, Germany, 2003.

[2] Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise, in: *Proc. 6th SIAM Int. Conf. Data Mining*, SIAM, Bethesda, Maryland, USA, 2006.

[3] Andrienko, N., Andrienko, G.: Spatial generalization and aggregation of massive movement data. *IEEE Trans. Visualization and Computer Graphics*, 17(2): 205-219, 2011.

[4] Bouguelia M.-R., Belaïd Y., Belaïd, A.: An Adaptive Incremental Clustering Method Based on the Growing Neural Gas Algorithm, in: *Proc. 2nd Int. Conf. Pattern Recognition Applications and Methods - ICPRAM 2013*, 42-49, 2013.